University of Saskatchewan
Department of Computer Science

CMPT 371.3
Systems Design, Implementation, and Maintenance

Final Examination
April, 2001

Closed Book

Cyril M. Coupal

Â Â Â Â Â Â Â Â
/150

Student Name: _____ Number: _____
Marks:Â Â Â Â Â 150 (about 1 minute per mark)

Please read the following carefully.

1. The examination is in two parts, 100 multiple choice questions followed by two long answer questions
2. Read all questions carefully. Make sure you understand what is asked
3. Do the easy questions first to make time for the more difficult questions
4. Do all questions. An unanswered question is worth zero
5. Write legibly. If I can not read it, I can not mark it.
6. Hand in all papers (multiple choice optical scan sheet, answer booklets and examination test booklet).
7. Do well.

Part ONE: There are 100 multiple choice questions, each worth 1 mark.

1. The three components of software: programs, documents, and data are collectively called:
   A. a module
   B. a configuration
   C. an architecture
   D. none of the above
2. It is possible to add people to a late software project, but the manager must be certain that:
   A. only the very best people are added
   B. the project is low risk
   C. the project is compartmentalized
   D. the project has gone no further than design
3. Software deterioration is due to side effects caused by
   A. corrections
   B. adaptations
   C. enhancements
   D. all of the above
   E. none of the above
4. The primary goal of software engineering is to:
   A. reduce the slope of the deterioration curve and the amplitude of the spikes on the curve
   B. improve the amount of documentation produced
   C. foster better communication among technical staff
   D. make projects more manageable
   E. none of the above
5. The term reusability refers to:
   A. the ability to reuse portions of code in other programs
   B. the ability to reuse standards and procedures from other projects
   C. the ability to perform object-oriented software development
   D. the ability to reuse program components
6. Software is business strategic because it:
   A. costs big bucks!
   B. represents high technology
   C. differentiates products and services
   D. directly results in more revenue
7. Which of the following is not one of the "layers" of software engineering?

A. tools
B. process
C. milestones
D. methods

8. Software engineering procedures encompass:
A. activities that help a manager plan, control, and track projects
B. methods that help a technologist build programs
C. a repository for CASE tools
D. none of the above

9. Which of the following activities properly fits in the process layer?
A. SQA
B. SCM
C. measurement
D. all of the above

10. Which activity is not part of analysis?
A. build models of the software that can be translated into design or code
B. help the developer to understand what the customer wants
C. model data, function and behavior
D. create algorithms that are the basis for code development

11. Why is design so important?
A. it serves as a basis for all steps that follow
B. it helps a developer to assess quality
C. it address architectural issues as well as algorithmic issues
D. all of the above

12. CASE tools can be integrated using:
A. a well-designed network
B. a well-designed repository
C. a single vendor tool set
D. all of the above
E. none of the above

Â

Chpt4

13. An organization that understands how to achieve quality will measure:
A. managers and practitioners
B. process and product
C. consistency and complexity
D. programs and data

14. Within the software engineering context:
A. a measure is the quantitative indication obtained by the act of measurement
B. a measurement is the quantitative indication obtained by the act of measuring
C. a metric is the qualitative measure of the degree to which a system possesses a given attribute
D. none of the above

15. When considered in the context of software metrics, normalization is used to:
A. avoid "apples and oranges" comparisons
B. simplify the database files that contain metrics
C. avoid the use of line of code measures
D. encourage the use of Cyclomatic complexity as a measure

16. Project data is normally collected:
A. throughout the entire software project
B. after the project has been completed
C. before the project begins
D. all of the above

17. Function points are a measure of software that:
A. indicate the functionality delivered by the system
B. indicate the number of inputs and outputs produced by the system
C. indicate the productivity of the system
D. indicate the number of user queries for the system

18. Which of the following metrics is an "after-the-fact" measure?
A. effort expended
B. overall cost
C. number of people involved
D. all of the above

19. Which of the following is not used in computing the function point value for a system?
A. number of internal data structures
B. number of inputs and outputs
C. number of system interfaces
D. number of user inquiries
E. number of data files

20. A feature point differs from a function point in the following way:
A. feature point values are more precise
B. feature point values consider complexity
C. feature point values consider the number of algorithms
D. feature point values consider the number of internal data structures

21. Backfiring is a technique that:
A. enables a manager to compute FP values using a best guess method
B. enables a manager to compute FP values using a LOC information
C. enables a manager to compute FP values only if third generation languages are used
D. none of the above

22. The average productivity of the typical software organization is in the following range:
A. 1 to 2 FP/person-month
B. 3 to 5 FP/person-month
C. 6 to12 FP/person-month

D. 13 to 20 FP/person-month

23. The most important factor that affects productivity and quality is:
A. the maturity of the software process
B. the SQA approach
C. the people
D. the amount of reusa

24. Technical metrics are used:
A. to judge quality after the software is delivered
B. to judge quality during design
C. to judge quality during testing
D. to judge quality throughout the software process

Chpt5

25. Reactive risk management is characterized by:
A. rapid reaction to all risks as they are identified during planning
B. rapid reaction only to important risks
C. crisis management
D. planned reaction to crisis-level arrors

26. The risk management paradigm is characterized by:
A. a linear process that identifies risk and acts on it
B. a process that prototypes risk
C. an evolutionary or iterative process
D. none of the above

27. Typical risk categories include:
A. non-critical and critical risks
B. project, technology, and business risks
C. trivial and non-trivial risks
D. human and machine risks

28. When performing risk identification, the manager should:
A. identify generic and project-specific risks
B. develop a risk management plan
C. estimate the cost of dealing with risk
D. all of the above

29. Risk projection includes:
A. an assessment of the probability that resources will be available to handle the risk
B. an assessment of the probability that the risk will occur
C. a projection of management's reaction to the risk
D. none of the above

30. Risk impact indicates:
A. the impact of defects on field use of the software
B. the impact of software failure on the safety of end-users
C. the impact of risk on the success of the project
D. the impact of risk on the development team

31. Which of the following is not a "risk component?
A. performance
B. cost
C. support
D. schedule
E. all are risk components

32. A refinement of the risk table can ba accomplished by replacing tha impact column with:
A. another probability column
B. a cost column
C. four columns that address the risk components
D. two columns that address levels of management concern

33. Risk mitigation focuses on:
A. understanding the risk
B. developing contingency plans should the risk occur
C. noting characteristics that may provide an indication that the risk will occur
D. avoiding the risk in the first place
E. none of the above

34. Risk monitoring focuses on:
A. understanding the risk
B. developing contingency plans should the risk occur
C. noting characteristics that may provide an indicaton thet the risk will occur
D. avoiding the risk in the first place
E. none of the above

35. Risk management focuses on:
A. understanding the risk
B. developing contingency plans should the risk occur
C. noting characteristics that may provide an indication that the risk will occur
D. avoiding the risk in the first place
E. none of the above

36. Once you've developed a risk table you should:
A. set it aside; it's served it's purpose by making you aware of risk
B. update the table regularly
C. update the table just before testing

chpt9

37. The first thing that happens as part of the change control process is:
A. a change request is received
B. the CCB makes a decision
C. the change is made
D. none of the above

38. A change request is:

A. given directly to the people who will make the change
B. evaluated by technical staff so that the CCB can make a decision
C. is completed by software engineers who assist the user
D. should always be transmitted electronically
E. none of the above

39. The software change request should be:
A. time stamped
B. prioritized
C. assigned a unique identifier
D. all of the above
E. none of the above

40. The software change report occurs as a consequence of:
A. evaluation of the requested change
B. the work conducted as the change is made
C. an SCM audit
D. an SQA checkpoint

41 The job of the CCB is to:
A. control the project associated with change
B. make a decision on whether a change should be made
C. audit the change process
D. produce the change report
E. none of the above

42. A CCB hierarchy is sometimes necessary because:
A. the scope of the change exceeds the scope of a CCB's authority
B. change occurs more rapidly when a hierarchy is in place
C. different constituencies can become involved
D. the CCB structure should mirror the organizational structure

43. Pushback occurs when:
A. the scope of the change exceeds the scope of a CCB's authority
B. when problems are difficult to fix
C. when the customer makes unreasonable demands for changes
D. none of the above

44. Check out helps an organization avoid:
A. SCIs that are too large
B. SCIs that are simultaneously changed by two or more people
C. SCIs that are undocumented
D. SCIs that reside in a supplementary repository

45. An engineering change order is:
A. the requirements specification for the change
B. the project plan for the engineering group
C. a written memo from the manager of software engineering
D. a document that accompanies a release of the software

46. A change is made by:
A. modifying the source code
B. updating all documents
C. following good software engineering practice
D. all of the above

47. Once the change is made, the following testing approach is common:
A. stress testing
B. regression testing
C. Cyclomatic testing
D. release cycle testing

48. Beta sites would be used as part of the release process when:
A. the change may impact hundreds or thousands of end users
B. the change is at alpha level
C. little qualification testing was conducted
D. none of the above

chpt13

49. To develop a stable software design, the first area of design focus should be:
A. algorithms
B. data
C. program structure
D. interfaces

50. Software design makes use of which of the following elements of the analysis model:
A. data model
B. functional model
C. behavioral model
D. all of the above
E. none of the above

51. Most people believe that design should be performed explicitly. This means that:
A. specific design models are created
B. there is explicit reference to code
C. explicit requirements are implemented
D. all of the above
E. none of the above

52. A data abstraction called STACK is to be implemented. It's reasonable to believe that corresponding procedural abstractions would exist. These would likely be:
A. push and pop
B. open and close
C. create and delete
D. push and pull

53. Stepwise refinement is a design approach that:
A. develops more detail with each iteration

B. develops less detail with each iteration
C. results in smaller computer programs
D. results in better data structures

54. Information hiding:
A. encodes all data within a module
B. constrains access to the internal details of a data structures or a module
C. is used only for data base design
D. is used only for procedural design

55. The primary benefit of information hiding is:
A. to improve the code
B. to shorten the program
C. to reduce unintended side effects
D. all of the above

56. When you design, the size of a module should be dictated by:
A. limitation on number of lines of code
B. its internal characteristics
C. the size of the module interface
D. the number of modules in the system

57. Two costs must be considered when selecting the right number of modules for an application (select from this list):
A. module development cost and integration cost
B. people costs and tools cost
C. integration cost and communication cost
D. none of the above

58. Cohesion indicates:
A. the degree to which data structures are contiguous across a system
B. the degree to which a module has been refined in a stepwise fashion
C. the degree to which a module performs a single, well-defined function
D. none of the above

59. To make the transfer of information across a large system as simple as possible, a designer specifies a large global data area that is accessible by all modules. How will this affect the coupling of the system?
A. It will improve coupling
B. it will result in high coupling
C. it will result in low coupling
D. it will have no effect on coupling

60. Design metrics can be used to measure which of the following software design characteristics
A. procedural complexity
B. architecture/structure
C. integration complexity
D. all of the above
E. none of the above

chpt17

61. Why can't we simply delay error removal until testing?
A. the error removal efficiency of testing is not high enough
B. testing methods aren't good enough
C. the testing strategy demands earlier removal
D. there's nothing wrong with waiting until testing

62. The primary objective of testing is:
A. to show that the program works
B. to provide a detailed indication of quality
C. to find errors
D. to protect the end-user

63. A software engineer should:
A. perform all tests because he/she knows the program best
B. perform some tests, but share responsibility with an independent tester
C. not be involved in testing
D. none of the above

64. Exhaustive testing is difficult or impossible because:
A. it would take too long
B. there are too many path permutations in most programs
C. it would be difficult to develop test cases to test every path permutation
D. all of the above

65. Selective testing is used to test:
A. every program path
B. selected independent paths
C. selected paths, data structures and variables
D. selected inputs and outputs

66. Tests that are conducted must be traceable to:
A. software requirements
B. performance constraints
C. hardware characteristics
D. all of the above

67. The test plan is developed:
A. just before testing commences
B. as soon as design is complete
C. as soon as requirements have been documented
D. during project planning

68. In general, a good test plan will assign testing effort in the following way:
A. evenly across all program modules
B. unevenly, focusing on the 40% that account for the majority of the errors
C. unevenly, focusing on the 20% that are likely to contain the majority of the errors

D. unevenly, with slightly more effort focused on these modules that are likely to be error prone

69. The best way to test a large program is:
A. test major chunks of the program to uncover errors
B. test incrementally
C. test sequentially
D. none of the above

70. Software testability is:
A. how easily [a computer program] can be tested
B. how adequately a particular set of tests will cover the product
C. how easily a program can be checked and repaired in the field
D. all of the above
E. none of the above

71. Testing should:
A. never be conducted outside normal operating conditions of the software
B. always be conducted outside normal operating conditions of the software
C. sometimes be conducted outside normal operating conditions of the software
D. testing outside normal operating conditions is impossible

72. Testing and debugging are:
A. different words that means essentially the same thing
B. related in that one uncovers errors and the other finds the cause
C. different activities that should always be performed by different people
D. both focused on symptoms

chpt8-1

73. Software exhibits quality if (pick the best answer):
A. it works
B. good documentation exists
C. it does what the customer wants it to do
D. there are few errors found during formal technical reviews

74. It is possible to measure quality, but:
A. it can only be measured indirectly
B. it isn't worth the enormous effort involved
C. only defect-related metrics can be collected
D. industry standards must be available

75. The McCall quality triangle stresses three areas:
A. requirements, design, and test
B. documentation, programs, and date
C. operation, revision, and transition
D. none of the above

76. Process-oriented measures are used to:
A. improve the manner in which data are processed
B. improve the software process itself
C. improve the structure of processing narratives
D. none of the above

77. Product-oriented metrics are:
A. collected in real time throughout the software engineering process
B. useful in determining customer satisfaction
C. part of every organization's software engineering approach
D. all of the above

78. Which of the following metrics is an "after- the-fact" measure:
A. effort expended
B. overall cost
C. number of people involved
D. all of the above

79. Which of the following is not a technical metric:
A. Cyclomatic complexity
B. code measures (e.g., Halstead)
C. defects/line of code
D. architectural complexity

80. Defect removal efficiency is defined as:
A. the number of errors found during reviews
B. a function of both errors and defects
C. a non dimensional number that has no upper bound
D. a percentage ranging from 0 to 75 percent

81. A software project manager contributes to product quality in the following way:
A. encouraging good customer-developer communication
B. performing good estimation
C. establishing mechanisms for change control
D. all of the above

82. A tester contributes to software quality:
A. by finding all errors in the software before it is released
B. by designing test cases with a high probability of finding errors
C. by demanding that every module be unit tested
D. none of the above

83. The following statement suggests how software requirements analysis and SQA are connected:
A. a good design leads to higher quality products
B. a written specification leads to high quality
C. analysis establishes a basis for determining conformance to requirements
D. the behavioral model is the key to high quality

84. How is quality assessed during software design?
A. using formal technical reviews
B. collecting technical metrics

C. applying effective design methods
D. all of the above

B5. Which of the following is not an objective or outcome of a formal technical review?
A. finding errors
B. determining compliance with project schedule
C. serving as a training ground
D. all are objectives of FTRs

86. The following review is the most formal of all FTRs:
A. walkthroughs
B. round-robin reviews
C. inspections
D peer reviews

87. The job of the review leader is to:
A. establish an agenda
B. control discussion that drifts
C. ascertain whether reviewers have prepared
D. all of the above
E. none of the above

88. How much preparation is recommended prior to the review?
A. 20-30 minutes
B. 1-2 hours
C. 45 minutes
D. preparation is not always necessary

89. An issues list is:
A. something prepared by each reviewer before the review
B. something prepared by the recorder during the review
C. something prepared by the producer during the review
D. something prepared the day after the review

90 Problem solving is something that:
A. should be encouraged during reviews
B should be avoided during reviews
C. should be conducted by a subgroup during the review
D. should be performed only by the review leader

91. The job of the recorder is to:
A. take detailed notes of all discussion during the review
B. make a list of all participants
C. record all action items
D. none of the above

92. The Technical Review Summary Report:
A. is produced monthly and summarizes all reviews conducted
B. is produced at the conclusion of each review
C. is produced by the manufacturer after all review of his/her products
D. is produced automatically using automated tools

93. When pointing out an error, a sensitive reviewer should:
A. ask a question
B. note it directly, without comment
C. tell the review leader before the review and ask the leader to mention it
D. not mention it until after the review

94. The following is a sign that a reviewer has not prepared:
A. asking lots of questions
B. reading from notes on the edge of the product pages
C. in-depth reading of the product pages
D. talking to a colleague

95. In general, the review team can come to *n* different decisions at the end of the review, where *n* is:
A. 2
B. 3
C. 4
D. the team need not come to a decision

96. The following is a primary difference between walkthroughs and inspections:
A. producer presents the product
B. reviewers prepare
C. written notes are taken throughout the review
D. a defined leader coordinates the review

My own questions

97. Before a project can begin, the manager and software team must:
A. estimate the work to be done
B. estimate the resources to be required
C. estimate the time that will elapse from start to finish
D. all of the above
E. none of the above

98. Software feasibility has four (4) solid dimensions:
A. abstraction, inheritance, aggregation, and association
B. documentation, testing, source code, executable code modules
C. technology, finance, time, and resources
D. all of the above
E. none of the above

99. When determining the scope of a project, the following must be evaluated together:
A. Client needs, wants and ability to pay
B. function, performance, and constraints
C. computer platform, software development environment, developer skills
D. all of the above

E. none of the above

100. To achieve reliable cost and effort estimates:

A. delay estimation until late in the project
B. base estimates on similar projects that have already been completed
C. use relatively simple decomposition techniques to generate project cost and effort estimates
D. all of the above
E. none of the above

Part TWO: Long answer questions.

[25] 101. Cyclomatic Testing.

Text Box: ALG. SOMEFUNCTION(A, B, C, D) RETURNS RSLT

```
RSLT ÃŸ 0
IF (A Ã³ 1)
THEN
I ÃŸ 1
WHILE (I Â£ A)
RSLT ÃŸ 0
IF (B > C)
THEN
RSLT ÃŸ RSLT + B
ELSE
IF (B < C)
THEN
RSLT ÃŸ RSLT + C
ELSE
RSLT ÃŸ 1
I ÃŸ I + 1
ELSE
IF (D < A)
THEN
RSLT ÃŸ 100
I ÃŸ 1
WHILE (I Â£ B)
RSLT ÃŸ RSLT + 1
I ÃŸ I + 1
ELSE
IF (D < C)
THEN
RSLT ÃŸ D
ELSE
RSLT ÃŸ C
RETURN(RSLT)
```

Consider the following algorithm:

a) [5 marks] Develop a flow graph for this algorithm.
b) [5 marks] Compute the Cyclomatic complexity of this algorithm.
c) [15 marks] For each basis path, give the following
Â· The path being considered
Â· The test values required to execute that path
Â· The expected result returned by the algorithm from that path

[25] 102. Consider the following UML logical view of the classification hierarchy for a smallÂ

project:

The light switch contains two control devices, an On/Off toggle and a dimmer control. The switch, in turn, controls a light bulb. The light bulb can be on or off and can have a given intensity from 0 to full wattage.

A motion detector is now being added to the system. The motion detector must meet the following requirements:

Â· It can have its power turned on or off
Â· It can be active (will sense motion) or passive (will not)
Â· The amount of sensitivity to motion can be controlled by a sliding control which either increases or decreases its sensitivity. Sensitivity can be from 0 to maximum sensitivity.
Â· If the motion detector becomes tripped (it has detected motion when active) it will flash the light bulb on and off until the sensor has been reset.

You must modify the design above so that the motion detector becomes part of the system. When remodeling the system, maximize reuse for later implementation and efficient